Sensors and Materials, Vol. 29, No. 7 (2017) 953–960
MYU Tokyo

S & M 1385

A Secure and Efficient Mutual Authentication Hand-off Protocol for Sensor Device Support in Internet of Things

Bruce Ndibanje, KiHwan Kim, YoungJin Kang,
HyunHo Kim, TaeYong Kim,[1] and HoonJae Lee[1*]

Department of Ubiquitous IT, Graduate School of Dongseo University, Sasang-Gu, Busan 617-716, Korea
[1]Division of Computer and Engineering, Dongseo University, Sasang-Gu, Busan 617-716, Korea

(Received April 8, 2016; accepted May 17, 2017)

Keywords: healthcare, IoT, sensor network, session key

To take advantage of the Internet of Things (IoT), it is essential that medical enterprises and the community should trust the IoT in systems in terms of performance, security, privacy, reliability, and return on investment, which are unaddressed challenges of current IoT systems. In this paper, we propose a secure and efficient mutual authentication protocol that supports IoT devices for handoff protocols. The proposed protocol is based on received signal strength (RSS) measurements and on public key cryptography and provides security against leakage resilience of private keys on untrustworthy networks. Performance analysis shows that the proposed scheme is efficient and resilient against various kinds of attacks.

1. Introduction

Healthcare systems related to the Internet of Things (IoT) are based on the essential definition of the IoT as a network of devices that connect directly with each other to capture and share vital data through a secure service layer (SSL) that connects to a central command and control server in the cloud. The idea of devices connecting directly with each other is, as the man who coined the term "Internet of Things" puts it, "a big deal".[(1)] Some recent work has tried to optimize the handover procedures (the mechanism for changing the access point of attachment of a node is known as handover procedure) for mobile solutions in an IoT. Hence, in this work, we have addressed the design of an optimal handoff procedure, building upon security and efficiency for a mutual authentication protocol to support the handoff process in IoT for healthcare applications.

2. Literature Review

Recently, He *et al.*[(2)] have introduced an interesting handover authentication protocol called PairHand. To improve the communication efficiency and reduce the burden on authentication server (AS), PairHand only requires two handshakes between a mobile node (MN) and an access point (AP) for mutual authentication and key establishment, instead of relying on the participation of AS. Furthermore, considering the high cost and the inconvenience of revoking users due to the use of a group signature in the authentication process, PairHand makes its construction directly

*Corresponding author: e-mail: hjlee@dongseo.ac.kr
http://dx.doi.org/10.18494/SAM.2017.1603

ISSN 0914-4935

based on pairing-based cryptography and uses a pool of shorter-lived pseudonyms to protect users' privacy. Unfortunately, shortly after this protocol was developed, He *et al.*[2] found that there was a serious design weakness in the PairHand protocol that enabled an adversary to easily obtain the private key from the message transported in the first round of the protocol. As a result, they presented an improvement by employing a composite order bilinear group, claiming that the improved version fixed the security problem without losing any of the desirable features of PairHand. At the same time, Tsai *et al.*[3] presented a handover authentication protocol considered to be secure, which solved the above security problem with PairHand but increased the size of the public key. Other recent work related to mobility management for IoT[4–6] has been proposed to secure data during the handover process.

3. Proposed Scheme for a Healthcare System in the IoT

3.1 Design and components architecture

As illustrated in Fig. 1, the design system architecture of our proposed protocol consists of three basics: MN, AP, and an AS. We consider a Healthcare Hospital with a ubiquitous sensor network, where medical staff can access data from their wireless mobile devices and patients with their nodes could move freely, passing from one AP's coverage area to another, thereby always ensuring their accessibility.

3.2 Initial authentication protocol

As described in Algorithm 1, this subsection gives details of the proposed protocol, and more details of how the algorithms are processed are given in the descriptions of the steps. As already mentioned in our list of assumptions, in this paper, we do not deal with the registration phases of all components in the healthcare wireless network. We assumed that the phase was already handled by the administrator of the network. The following are the steps of the initial authentication protocol issued when the MN wants to access the network for the first time.

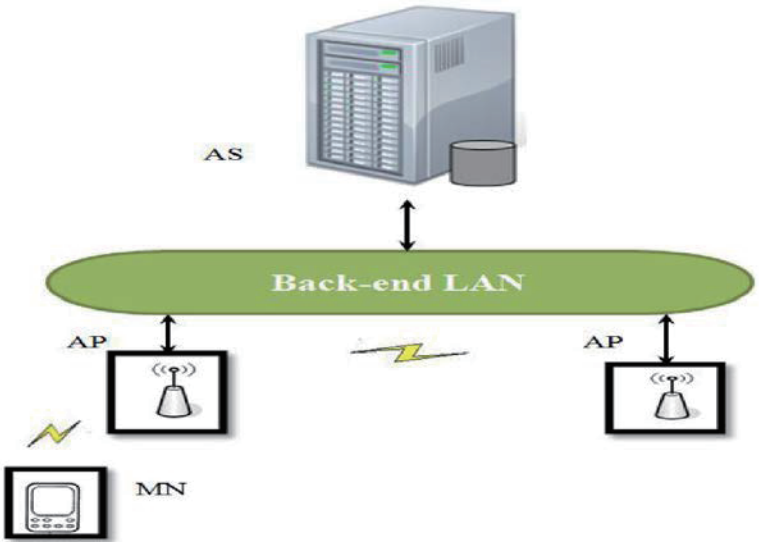

Fig. 1. (Color online) Design and components.

Algorithm 1: Initial Authentication Protocol

// **Actions triggered by MN**
1: Compute MNAuth⟵ h(IDMN||MACMN)
2: Broadcast to APcurr: {ReqAuthMssg}//Session initialization
// **Actions executed by APcurr**
3: Verify if MNAuth *≠ MNAuth
4: **if yes, abort.**
5: **Otherwise,**
6: Compute APResp= h(IDAPcurr||IDMN||MACMN)
7: Broadcast to AS: { RqAuthSvc}
//**Actions executed by AS**
8: **Check if (TAS− TAP) ≤ ΔT**
9: **Else if** APResp = APResp *,
10: **If yes,**
11: Compute KSES=(IDMN||IDAPcurr||IDSAS)
12: Compute AccpMsgSvc= EKSES(IDAS)
13: Broadcast to AP : {AccpMsgSvc}
//**Actions executed by APcurr**
14: Decryption: DKAPcurr {AckAccpSvc}.
15. **Check if (TAP− TAS) ≤ ΔT**
16: **Else if** IDAS=IDAS*, IDAPcurr= IDAPcurr** If yes,
17: Compute AckAccpSvc = EKSES (IDAPcurr)
18: Broadcast to MN: { AckAccpSvc }
//**Actions executed by MN**
19: Decryption : DKMN {AckAccpSvc}
20: **Verify if** IDMN = IDMN**, MACMN = MACMN**,
21: **Else if** IDAPcurr = IDAPcurr**
22: **If Yes**, MN believes AP is real,
23: **Otherwise**,
24: **Return false** and **abort**.
25: **end if**
26: **end if**
27: **end if**
28: **end if**
29: **end if**
30: **end if**
31: **end if**
32: **end if**
33: **end if**
34: **end if**
35: **end if**

Step 1: MN sends to the nearest current AP a MAC-based authentication message. We assume that all MAC addresses of all allowed devices are preconfigured into all APs. The MN computes the following: MN compute $MNAuth = h(IDMN||MACMN)$) to current AP.

$$MN\ sends\ to\ APcurr\text{: }\{ReqAuthMssg\} \tag{M1}$$

Step 2: The current AP checks the received hash value; if it matches the preregistered value, then the MN is on the allowed list and the current AP accepts the authentication request. Otherwise, the request is rejected, and the message is forwarded to the AS to request network access. The current AP prepares the request message to the network and

computes the following: $RqAccNet = h(IDAPcurr||IDMN||MACMN)$.

$$APcurr\ sends\ to\ AS: \{RqAccNet\} \quad \text{(M2)}$$

Step 3: When the AS receives the message from the AP, it validates the time *TAS* and checks if $(TAS–TAP) \leq \Delta T$; if yes, then it continues to verify the hashed value; if it matches the one preregistered, then it goes to the next step. Otherwise, it rejects it. After the verification and validation process is finished, the AS informs the current AP that the MN and the user are legitimate. The AS computes the session key and generates a secret key *KAS* and encrypts the acceptance message. The AS prepares a message of acceptance of network service to the AP and also sends a session key.

$$KSES = h\,(IDMN||IDAPcurr||IDAS)\ and\ \ AccpMsgSvc = EKSES\,(IDAS)$$
$$AS\ replies\ to\ APcurr: \{AccpMsgSvc\} \quad \text{(M3)}$$

Step 4: Upon receiving the acceptance message from the AS, the current AP decrypts it and verifies if the AS is legitimate, and it validates the time; if it is legitimate, the process is aborted. If $(TAP - TAS) \leq \Delta T$ is yes, then it continues to the next step. Otherwise, the process is stopped. *DKMN* {*AckAccpSvc*} and mutual authentication: $IDAS = IDAS^{*}$, $IDAPcurr = IDAPcurr^{**}$. After mutual authentication is complete, the current AP sends an acknowledgment message to the MN along with its id encrypted by the session key.

$$AckAccpSvc = EKSES\,(IDAPcurr)$$
$$APcurr\ replies\ to\ MN: \{AckAccpSvc\} \quad \text{(M4)}$$

Step 5: While receiving the acknowledgment message from the current AP, the MN performs mutual authentication by verifying some secret parameters. After the decryption of the arrived message, *DKMN* {*AckAccpSvc*}, the MN stores the *IDAS* and verifies the following:

$$IDMN = IDMN^{**},\ MACMN = MACMN^{**},\ IDAPcurr = IDAPcurr^{**} \quad \text{(M5)}$$

If yes, then the MN believes that the current AP is real; otherwise, not. Now, the MN has the right to join the network and the user can access data using mobile devices. Moreover, the involved entities share the symmetric session key $KSES = h(IDMN||IDAPcurr||IDSAS)$ for performing further subsequent operations during a session.

3.3 Handover process and received signal strength (RSS) computation

Algorithm 2 describes how the handover protocol works; it is invoked when the user wants to switch to a new AP from the current AP. Before the handover starts, the device algorithm analyzes the handover initiation between the 3G and wireless local area networks (WLANs) in the case of cell overlapping as described in Fig. 2.

Algorithm 2 : *Handover Process*

// Actions triggered by MN
1: Compute *MNHdv = h(IDMN⊕MACMN)*
2: Broadcast { *HdvRqMsg*} to *APNew*
// Actions executed by *APNew*
3: **Verify if** *MNHdv ≠ MNHdv* *
4: **If yes**, a**bort**
5: **Otherwise**,
6: Compute *Hd=h(IDAPNew||IDMN)*
7: Broadcast *{HdvAccMsg}* to MN
//Actions executed by MN
8: **Verify if** *Hd = Hd* *
9: **If yes**, t**hen**,
10: *APNew* is legal entity // MN continue to use the network
11: **If not**,
12: **Return false** and **abort**.
13: **end if**
14: **end if**
15: **end if**
16: **end if**
17: **end if**

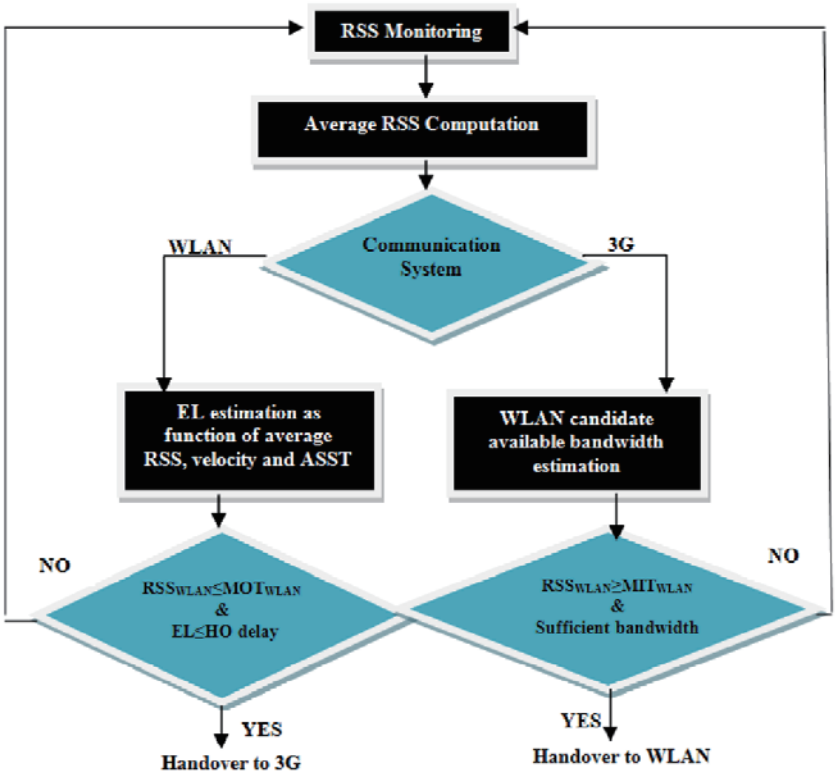

Fig. 2. (Color online) RSS monitoring and computation procedures

Step 1: The MN first computes the strength of the received signal of the current AP. The signal strength level (*SSth*) of the required value of RSS is set to initiate the handover. Thus, when the value of RSS (the value is set below *SSth*, *RSS*) of the current AP drops below the *SSth*, the handoff is triggered before the MN moves beyond the coverage area of the current AP. This value can also estimate the distance between devices. If the mean received signal strength drops below the *SSth,* then the handoff should be executed. The calculation of the RSS value is given by Eq. (1) as described in Ref. 7, where *N* is the measured RSS value and *RSSi* represents each single measurement.

$$\overline{RSS} = \frac{1}{N}\sum_{1}^{N} RSSi \tag{1}$$

Step 2: After triggering the handover process, the authentication protocol now takes place from the current AP with $RSS < SSth$ to a new AP with a strong RSS signal. The MN then sends the request handover message to the new AP with the following: MN computes $MNHdv = h\,(IDMN \oplus MACMN)$; the MN sends the *HdvRqMsg* to APNew.

$$MN\ sends\ to\ APNew\text{: } \{HdvRqMsg\} \tag{H1}$$

Step 3: The new AP decrypts the message and validates the time, checking the following: $MNHdv = MNHdv^*$; if yes, it goes to the next step. Otherwise, abort. The new AP computes $Hd = h(IDAPNew \| IDMN \| MACMN)$ and sends the handover access message to the MN.

$$APNew\ replies\ to\ MN\text{: } \{HdvAccMsg\} \tag{H2}$$

Step 4: While receiving the access handover message, the MN decrypts the message and validates the time and checks if the hashed value matches the stored one or not: $Hd = Hd^*$; if yes, it continues to have network service. Otherwise, abort.

4. Performance Analysis

4.1 Security services analysis

1. *Masquerade mobile node attacks*: The protocol works against this attack in concept. We assume that an attacker wants to access the network using a mobile device. In this case, the attacker will attempt to connect his device by sending the {*ReqAuthMssg*}. Without any problem, the current AP will reject the request because it is an unknown device (id and MAC's of the device are unregistered).
2. *Source substitution attack*: This attacker can use another entity's public key and manipulate it to obtain a certificate in the name of the attacker for the value of that public key. Thus, it permits the attacker to impersonate the system to be a legitimate user and the signer of data. Similar to this concept, an attacker can intercept {*ReqAuthMssg*} and then reuse the certificate claiming to be the legitimate owner. In this scenario, the system will compute all parameters and check if the device is registered, which is true.

Table 1
Communication overhead.

Messages	Length (bits)
MNAuth ⟵ h(IDMN||MACMN)	128
APResp = h(IDAPcurr||IDMN||MACMN)	192
KSES = (IDMN||IDAPcurr||IDAS)	192
AccpMsgSvc = EKSES(IDAS)	256
AckAccpSvc = EKSES (IDAPcurr)	256
MNHdv = h(IDMN||MACMN)	128
Hd = h(IDAPNew||IDMN)	128
Total length of all messages	1290

Table 2
Comparison with existing schemes.

Scheme	Communication overhead (bits)
Santanu *et al.*[7]	1400
Huifang *et al.*[8]	3168
This work	1290

3. *Session key establishment*: A session key, *KSES*, is established between the communicating entities after the authentication process. This key is different in each session and cannot be replayed after the session expires.

4.2 Efficiency analysis: communication overhead

Table 1 gives the numerical results and shows that the proposed protocol is efficient in terms of memory usage. Table 2 compares existing schemes from Refs. 5 and 6. In this study, all ids and location ids are 64 bits in length, while the keys are 128 bits long and secret parameters and nonces are 32 bits in length. We also assigned to each message a size of 64 bits, which includes a protocol id, message id, and other packet format data. Then, after the calculation of all messages, our protocol requires a total length of 1545 bits. For more details, we can see that, in Algorithm 1, the initial authentication process requires only 4 messages (Steps 1 to 4) to achieve the mutual authentication process and session key generation requiring a length of 1034 bits. In the case of Algorithm 2, the handover process requires just 2 messages (Steps 1 to 2) requiring 256 bits. In this work, overhead analysis takes into considerations mica2 motes as sensor nodes. It is obvious that the cost of communication is inexpensive for a given wireless node device.

5. Conclusions

In this study, we proposed a secure and efficient approach to designing a mutual authentication protocol to support handover processes applied to healthcare systems using wireless devices in the IoT configuration. As part of the cryptographic function, we have described another vital function (signal strength) to trigger the handover process. In addition, this is a fast protocol because the AS is not involved in the handover process, and the access point and AS are connected via a back-end LAN, which is a wired network operating at a very high speed. Furthermore, the results of the analysis of efficiency and security reveal that the proposed scheme is efficient and resilient to various attacks.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number: NRF-2016R1D1A1B01011908). It is also supported by the BB21 project of Busan Metropolitan City.

References

1 Gartner‘s Hype Cycle Special Report for 2015, Gartner Inc. (2015): http://www.gartner.com/technology/research/hype-cycles/ (accessed July 2015).
2 D. He, J. Bu, S. Chan, and C. Chen: IEEE Trans. Wireless Commun. **11** (2012) 48.
3 J. Tsai, N. Lo, and T. Wu: Wireless Pers. Commun. **73** (2013) 1037.
4 H.-S. Chai, J.-Y. Choi, and J.-P. Jeong: An Enhanced Secure Mobility Management Scheme for Building IoT Applications. International Workshop on Networking Algorithms and Technologies for IoT (NAT-IoT 2015).
5 S. M. Ghaleb, S. Subramaniam, Z. A. Zukamain, and A. Muhamed: EURASIP JWCN (2016) 165.
6 S.-M. Chun, H.-S. Kim, and J.-T. Park: Sensors **15** (2015) 16060.
7 C. Santanu, A. K. Da, and J. K. Sing: J KSU–CIS **26** (2014) 181.
8 Huifang, G. Linlin, and X. Lie: Sensors **15** (2015) 17057.